Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-30. (canceled)

31. (currently amended) A pipelined microprocessor comprising:

an instruction cache that is indexed by a fetch address, said instruction cache for caching instructions, and for providing said instructions to an instruction buffer for storage therein, wherein said instructions comprises variable byte-length instructions;

a branch target address cache, coupled to said instruction buffer and indexed by said fetch address, for caching branch target addresses of previously executed branch instructions;

said instruction buffer comprising an indicator associated with each byte of each of said instructions stored in said instruction buffer ~~at least one indicator associated with each of said instructions~~, wherein said indicator has a true value if said branch target address cache predicts that said byte is an opcode byte of said instruction and that said instruction is one of said previously executed branch instructions and the microprocessor has speculatively branched to one of said branch target addresses cached for said one of said previously executed branch instructions.

32. (canceled)

33. (canceled)

34. (currently amended) A method of speculatively branching in a pipelined microprocessor, comprising:

caching, in a branch target address cache (BTAC), a plurality of branch target addresses of previously executed branch instructions and a bit associated with each of said branch instructions, wherein said bit is true only if ~~direct indication of whether each of~~ the associated branch instruction[[s]] spans more than one instruction cache line;

accessing said BTAC with a fetch address of an instruction cache after said caching;

determining whether said fetch address hits in said BTAC in response to said accessing; and

branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address.

35. (original) The method of claim 34, further comprising:

storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC.

36. (original) The method of claim 35, wherein said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken.

37. (currently amended) The method of claim 34, wherein the microprocessor processes variable byte-length instructions, the method further comprising:

storing in an instruction buffer instructions provided by said instruction cache selected by said fetch address, and storing a discrete indication for each byte of each said instruction, wherein said discrete indication is true if said BTAC predicts said byte is an opcode byte of said instruction and said branching the microprocessor to one of said plurality of branch target addresses was performed for said instruction ~~that said branching was performed if said branching is performed; and~~

~~determining from said discrete indication, subsequent to said storing, that said branching was performed~~.

38. (currently amended) The method of claim 37, further comprising:

determining from said discrete indication, subsequent to said storing, that said branching was performed~~wherein said storing said indication said branching was performed comprises storing said indication in an instruction buffer~~.

39-42. (canceled)

43. (currently amended) The microprocessor of claim ~~32~~ 31, further comprising:

instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte; and

prediction check logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein if one of said ~~plurality of~~ indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if ~~a same~~ said byte of said one of said instructions associated with said one of said indicators is not indicated by said instruction decode logic ~~and by said plurality of indicators stored in said instruction buffer~~ to be said opcode byte.

44. (currently amended) The microprocessor of claim 31, further comprising:

instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate whether each of said instructions is a non-branch instruction; and

prediction check logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein if one of said ~~plurality of~~ indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said instruction decode logic indicates said one of said instructions is a non-branch instruction.

45. (currently amended) The microprocessor of claim 31, wherein said branch target address cache is further configured to cache a length of each of said previously executed branch instructions, the microprocessor further comprising:

instruction decode logic, coupled to said instruction buffer, for determining a length of each of said instructions; and

prediction check logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein if one of said ~~plurality of~~ indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said previously executed branch instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said length received from said instruction decode logic does not match [[a]] said ~~speculative~~ length of said one of said previously executed branch instructions provided by said branch target address cache.

46. (currently amended) The microprocessor of claim 31, further comprising:

prediction check logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said ~~plurality of~~ indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved direction of said branch instruction does not match a direction of said branch instruction predicted by said branch target address cache.

47. (currently amended) The microprocessor of claim 31, further comprising:

prediction check logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said ~~plurality of~~ indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved target address of said branch instruction does not match said one of said branch target addresses to which the microprocessor speculatively branched.

48. (currently amended) The microprocessor of claim 31, further comprising:

a non-speculative branch predictor, coupled to said instruction buffer, for generating a non-speculative predicted target address of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched; and

branch control logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein if one of said ~~plurality of~~ indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to said non-speculative predicted target address if said non-speculative predicted target address generated by said non-speculative branch predictor does not match said one of said branch target addresses of said branch instruction provided by said branch target address cache.

49. (currently amended) The microprocessor of claim 31, further comprising:

a non-speculative branch predictor, coupled to said instruction buffer, for generating a non-speculative predicted direction of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched; and

branch control logic, coupled to receive said ~~plurality of~~ indicator[[s]] associated with each byte of said instructions from said instruction buffer, wherein if one of said ~~plurality of~~ indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to a next instruction sequential to said branch instruction if said non-speculative predicted direction generated by said non-speculative branch predictor is a not taken prediction.

50. (currently amended) A branch target address cache (BTAC) for providing a speculative target address to address selection logic, the address selection logic selecting a fetch address for addressing a line in an instruction cache, the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache line, the BTAC comprising:

an array of storage elements, configured to cache target addresses of previously executed branch instructions and to store speculative branch information associated with said previously executed branch instructions, wherein said speculative branch information comprises a bit associated with each of said branch instructions, wherein said bit is true only if ~~direct indication of whether~~ the associated branch instruction presumed present in the cache line spans more than one line in the instruction cache;

an input, coupled to said array, for receiving the fetch address, to index into said array of storage elements to select one of said target addresses; and

an output, coupled to said array, for providing said one of said target addresses indexed by the fetch address to the address selection logic;

wherein said output provides said one of said target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address.

51. (previously presented) The branch target address cache of claim 50, further comprising:

a second output, coupled to said array, for providing a portion of said speculative branch information to control logic for controlling the address selection logic in response to said portion of said speculative branch information.

52. (previously presented) The branch target address cache of claim 50, wherein said speculative branch information comprises information predicting whether the branch instruction presumed present in the cache line will be taken.

53. (previously presented) The branch target address cache of claim 52, wherein said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit.

54. (previously presented) The branch target address cache of claim 52, wherein said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits.

55. (previously presented) The branch target address cache of claim 54, wherein said plurality of bits is stored in a saturating up/down counter.

56. (previously presented) The branch target address cache of claim 51, wherein said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address.

57. (previously presented) The branch target address cache of claim 56, wherein said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved.

58. (previously presented) The branch target address cache of claim 56, wherein said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output.

59. (previously presented) The branch target address cache of claim 50, wherein said speculative branch information comprises information specifying a location within the cache line of the branch instruction presumed present in the cache line.

60. (previously presented) The branch target address cache of claim 50, wherein said speculative branch information comprises a length of the branch instruction presumed present in the cache line.

61. (previously presented) The branch target address cache of claim 50, wherein said speculative branch information comprises an indication of a type of the branch instruction presumed present in the cache line.

62. (previously presented) The branch target address cache of claim 61, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a call instruction.

63. (previously presented) The branch target address cache of claim 61, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a return instruction.

64. (previously presented) The branch target address cache of claim 50, wherein each of said storage elements is configured to cache a plurality of target addresses.

65. (previously presented) The branch target address cache of claim 50, wherein the branch target address cache is external to the instruction cache.

66-70. (canceled)